chi+med
making medical devices safer

EPSRC Programme Grant EP/G059063/1

Public Paper no. 73

Supporting Field Investigators with PVS: A Case Study in the Healthcare Domain

Paolo Masci, Dominic Furniss, Paul Curzon,
Michael Harrison & Ann Blandford

Masci, P., Furniss, D., Curzon, P., Harrison, M. D. & Blandford, A. E. (2012). Supporting field investigators with PVS: A case study in the healthcare domain. *Proceedings of 4th International Workshop on Software Engineering for Resilient Systems (SERENE 2012)*, 150–164. Lecture Notes in Computer Science, vol. 7527. Springer.

PP release date: 12 July 2012

file: WP073.pdf

UCL UCLIC Swansea University Prifysgol Abertawe FITLAB Queen Mary University of London CITY UNIVERSITY LONDON EPSRC Engineering and Physical Sciences Research Council

Supporting field investigators with PVS: a case study in the healthcare domain

Paolo Masci[1]*, Dominic Furniss[2],
Paul Curzon[1], Michael D. Harrison[1], and Ann Blandford[2]

[1] School of Electronic Engineering and Computer Science
Queen Mary University of London
{paolo.masci,paul.curzon,michael.harrison}@eecs.qmul.ac.uk

[2] UCL Interaction Centre
University College London
{d.furniss,a.blandford}@ucl.ac.uk

Abstract. This paper reports the lessons learnt about the benefits of using a formal verification tool like PVS to support field studies. The presentation is based on a field study in the healthcare domain which was designed to investigate the resilience of human behaviour in an oncology ward of a hospital. The automated reasoning tool PVS was used systematically to compare actual practice observed during the field study with normative behaviour described for example by user manuals for the devices involved. The approach helped (i) identify latent situations that could lead to hazard, and (ii) suggest situations likely to warrant further investigation as part of the field study. The main contribution of this paper is a set of detailed examples that illustrate how we used PVS during the field study, and how the tool led to insights.

Keywords: Experience report; Field study; Socio-technical system; Automated reasoning; PVS.

1 Introduction

One approach to understanding complex socio-technical systems prior to introducing new technology is to undertake field studies. Field studies involve going to the real workplace to investigate how work is actually done, e.g., by observing workers and asking them questions. Little research addresses tool support that would help investigators during these field studies. To date, software tools developed and used by field researchers mainly focus on storing and encoding information — see for instance [14]. This paper explores how an already existing tool typically used to verify hardware and software can be used to reason about the field study data that is collected in a way that is accessible to field researchers. The aim is to automate routine checks, for example how consistent the information is, as well as to identify systematically those situations that

* Corresponding author.

are likely to warrant further investigation. The concern is to identify situations where gaps in the way artefacts and information resources support user actions may create sufficient preconditions for unsafe user actions potentially leading to harm. Previous work [2,6,7] has explored the benefits of using verification tools when re-analysing data from already completed field studies. In particular, we were able to gather additional insights about the socio-technical system, and we argued there that such an analysis could help when performed *during* the field study. However we had no direct evidence to support the claim. This paper builds on that research, in the context of a live field investigation. We report on our experience and the lessons learnt.

Contribution. We illustrate in detail how we used the PVS verification system to support investigations as part of a field study. The focus relates to specific situations faced during a field study carried out in the oncology ward of a hospital. Relatively simple use of the PVS verification system (i) helped identify latent situations that could lead to hazard, and (ii) suggested situations that warranted further investigation as part of the field study.

The rest of the paper is structured as follows. Section 2 makes clear how field studies and formal methods complement each other, and the benefits that can be obtained from their combined use. Section 3 describes artefacts and technologies involved in a particular procedures relating to glucose monitoring that were considered during the field study. Section 4 briefly introduces the PVS verification system and describes the modelling approach and the developed PVS models. Section 5 illustrates how a relatively simple use of PVS can provide useful insights for the field study. Specific real situations we faced during the field study in the oncology ward of a hospital will be the focus. They illustrate the obtained results and the lessons learnt. Section 6 contrasts and compares the approach with related work in the area and draws conclusions.

2 Integrating field studies and formal methods

Formal methods and field studies are traditionally seen as alternative approaches to analysing a system derived from completely different paradigms. On the one hand, formal methods are typically used to verify whether given properties hold for a model of the system under certain assumptions, and they are typically concerned with normative behaviour described in manuals and protocols. On the other hand, field studies focus on empirical approaches and aim to analyse how a system works 'in the wild' when deployed in the real world.

An alternative and more constructive perspective is to see the two approaches as complementary [15], and use them within a cyclical process where they feed each other. This paper illustrates the lessons learnt from a successful story of integration of field studies and formal methods to analyse the resilience of a complex socio-technical system — a hospital ward where a new device has been introduced with the aim of making the overall system more robust and efficient.

The field study proved invaluable in that it enabled: (i) identification of the information resources required by participants in the work; (ii) understanding of the difference between work as intended and documented as procedures, and work as practised; (iii) understanding of the gaps in the way that actions are resourced; (iv) a broad exploration of safety goals that are not achievable because circumstances conspire against them.

The formal reasoning tool has been used at the same time as field study to inform it and be informed by it. It provided essential benefits in: (i) making the categories precise; (ii) analysing the links between information resources and actions systematically; (iii) exploring the consequences of prescribed procedures systematically.

The rest of this paper develops these points and illustrates in detail our pragmatic use of field study data and formal methods.

3 Glucose monitoring procedure in the oncology ward of a hospital

This section illustrates some relevant observations made in the field about the actual use of the introduced technology in the oncology ward under investigation. The technology adopted is a handheld wireless device to help clinicians get frequent blood glucose readings from patients. The field study was performed during the first months of introduction of the device.

The ward has 24 beds for accommodating patients that need to stay overnight in the hospital whilst they are treated. Whilst in hospital, patients who are diabetic need their blood glucose levels managed. These levels can be affected by the patient's glucose intake, treatment and condition, so they need to be monitored closely to make sure they are not too high or too low. Either could lead to further health problems. The introduced device is used to monitor these levels. Information recorded on the meter during a reading can be automatically stored in the patient's record.

The main features of the device and the procedure described in the manual to perform the blood glucose test are the focus of attention in what follows. This information is compared with the procedures followed by clinicians observed during the field study. The description we give here is provided at a level of detail that is adequate to illustrate the benefits of using the verification tool during a field study. These descriptions are translated into a PVS model (Section 4) so that PVS can be used to analyse the model (Section 5).

3.1 Blood glucose meter and accessory box

The blood glucose meter adopted is a palm-sized portable device [9] that allows clinicians to measure blood glucose levels by means of small test strips that are inserted in a slot at the top of the device. The device has a touch screen that can be used by clinicians to view the patient's previous blood glucose test results as well as the patient's record. A reproduction of the device layout is shown in

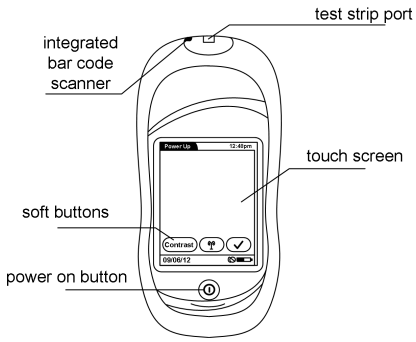

Fig. 1. Reproduction of the blood glucose meter

Figure 1. The device also has an integrated barcode scanner located close to the port that receives test strips used for identification purposes. The device supports a number of features, which we summarise.

Clinician and patient identification The device has a built-in barcode scanner that allows a clinician to identify a patient as well as the clinician taking the reading. This feature aims to guarantee that (i) only authorised personnel can use the device, and (ii) blood glucose readings are automatically associated with the right patient.

Wireless connectivity The device can be configured to communicate with a central server through a wireless connection. This makes it possible to synchronise in real-time with a central data management system that contains information about patients.

Quality control Device functionalities and accuracy must be checked regularly using special liquid compounds ("control and linearity solutions") that make sure that the glucose testing is accurate and reliable.

The glucose meter ensemble includes an accessory box for convenient transportation of meter and consumables. The accessory box contains: two test strip vials to perform the test, single use lancets to puncture the patient's finger, small boxes for disposable white swabs, control and linearity solutions to perform quality control tests. The glucose meter is battery-powered. A base unit is provided to recharge the meter's batteries when docked in the unit. The unit can be hardwired to a local area network to enable data transfer to and from a central data management system.

3.2 Normative procedure from the user manual

The normative procedure described in the device user manual [10] and various training material [13] is given as a sequence of steps. The procedure is the following (we adapted the original text here to match the level of detail needed for our illustrative example).

1. **Identify clinician** Power on the device and scan the clinician's badge.
2. **Identify patient** Enter the patient's account number by either scanning the patient's bar coded wristband or manually.
3. **Verify test strip code** Verify the strip code information by pressing "scan" and scanning the vial barcode.
4. **Insert test strip in the meter** Insert a test strip with the test strip window facing up. Insert the end with the silver bars. Insert test strip before dosing.
5. **Obtain blood sample** When the flashing drop icon appears on the meter display, obtain a blood sample
6. **Wait for the results** An hourglass will appear on the display while waiting for the result. When the results are ready, choose an appropriate comment from a list of pre-loaded comments, as necessary.
7. **Remove test strip from the meter** Remove the test strip from the meter and discard it according to infection control policy.
8. **Turn off meter and dispose gloves** Press the power off button to turn the meter off. Remove gloves and dispose of them.
9. **Document test results** Document the blood glucose result with date, time, and clinician's initials on flow sheet or chart as required.

3.3 Actual procedure observed at the hospital ward

The field investigator provided the following notes after spending some days on the ward. He observed various clinicians during their visits to patients for the blood glucose test, and talked to them to obtain explanations of what they did. In the following we report some relevant observations (we adapted the original text from the field notes to match the level of detail needed for our illustrative example).

Prepare trolley The blood glucose test was performed on a series of patients. Because of this, the clinician prepared a wheeled trolley to carry all the necessary equipment. The wheeled trolley contained two platforms to separate clean apparatus from waste. The top platform was used to hold the accessory box and a box of disposable gloves. The bottom platform had a bin for sharps and special refuse. A cardboard tray was also carried on the trolley for temporary location of used items, e.g., used swabs or used test strips, before they were transferred to the bin for sharps and special refuse.

Annotate bed numbers The numbers of the beds that are to be visited are reported on handover sheets. The clinician scribbled these bed numbers on the cardboard tray so as to create a checklist that was easy to access and update.

Unlock device The glucose meter is undocked from the base unit, powered on, and unlocked to initiate a new session. This is done to check whether the device is functioning properly and whether the device requires a quality control test. The badge number of a trained member of staff is needed to unlock the device and initiate the session. Such a number can either be scanned or manually entered.

Visit patients The visit starts after checking that the trolley and the accessory box has sufficient consumables for the whole set of patients that must be visited. At each bed, the clinician carried out the following sequence of activities:

- Ask the patient consent to perform the test. Some patients may refuse to do the test, or may be away from their bed for other reasons, e.g. treatment in other areas of the hospital. In these cases, the clinician skips the patient and goes to the next bed in the list.
- Scan the patient barcode. If the device is not able to correctly scan the barcode (e.g., because the wristband is crumpled), the clinician enters the number manually. Sometimes the device does not have information about the patient; in these cases, the device allows the clinician to continue and they proceed.
- Scan the strip container.
- Puncture finger.
- Insert test strip in the device (this activity is sometimes performed before puncturing the finger).
- Put blood on the strip.
- Wait for the test results.
- Record the test results in the glucose chart. The chart includes a number (the current test result), and a graph reporting the history of results (the clinician updates the graph with the current result). If the result is too high or too low, the clinician must report this to the nurse looking after that patient.

Visit end When all patients have been visited, the device is docked to the base unit and the trolley emptied.

4 PVS models

In this section, we illustrate the PVS models we developed out of the gathered field study data. The field data were organised according to the DiCoT method [3]. This method encodes relevant categories that are employed by the various users within the system and their means of communication. Inevitably the field data are open to alternative interpretations. The advantage of a less formal approach like DiCoT is that it enables an appropriate representation of a complex and rich system. However in order to assess the adequacy of the resources in supporting the actions needed to achieve goals, the PVS models go a stage further to make precise certain assumptions. These models can be seen as hypotheses about the meaning of the information resources within the activities. As hypotheses they are open to alternative interpretations that may have different more or less consistent implications. A PVS model is therefore not intended to be definitive and a number of alternatives could be developed in order to explore different assumptions about the circumstances.

4.1 Modelling approach

The modelling approach used has been outlined and illustrated in the context of other field studies including incident analysis in [6,7,5,8]. It involves the following steps: (1) modelling information resources reported in the field study data (e.g., glucose meter displays, information on staff badges, content of the accessory box); (2) modelling how information resources propagate within the system (e.g., how a staff ID is entered into the glucose meter); (3) formulating and verifying conjectures about how resources were used (e.g., were relevant resources available at critical moments to relevant actors) and facts about the prescribed use of information resources (e.g., according to procedures and regulations). The first two steps of the procedure aim to help field investigators externalise facts about what information resources are available to users and how such resources are used. The third step aims to check whether information resources provide constraints that are correct and tight enough to support safe user actions.

4.2 Modelling language

PVS models are specified in a strongly typed higher-order logic, which allows quantification over propositional functions to be formulated. The language includes the usual base types (e.g., `bool`, `nat`, `integer` and `real`), function type constructors `[A -> B]` (predicates are functions with range type `bool`), and abstract data types. Models are packaged in modular components called *theories*, which can be parametric in types and constants. They can use definitions and theorems of other theories by importing them. A language mechanism used extensively in the models considered here is *predicate subtyping* [12], which makes it possible to express complex consistency constraints. When using expressions with subtypes, PVS automatically generates proof obligations, called type correctness conditions (TCCs), that ensure the valid use of the type. We will rely on this automatic generation of proof obligations to check various consistency constraints for the use of information resources.

4.3 Developed models

We model artefacts and information resources used by clinicians during the blood glucose monitoring procedure. The models aim to enable a systematic analysis of consistency of collected data and safety checks. The concern is to identify situations where gaps in the way artefacts and information resources support user actions may create sufficient preconditions for unsafe user actions potentially leading to harm, such as wrong patient identification or unsafe disposal of refuse.

In the developed models, each information resource and artefact is specified using a different PVS datatype. The level of detail used in the PVS models was discussed with the investigator in relation to their findings as it must be consistent with the level of detail understood by the investigator. The aim is to create a model that embeds the facts observed by the investigator during the field study in a precise way, but without the need to be too specific when

certain information is not available or not deemed relevant by the investigator. In the developed models, the collection of information resources and artefacts represents the state of the system. The use of information resources and artefacts is modelled as transition functions over system states. This will be illustrated by demonstrating some relevant features of the specification of the models. The complete PVS models relative to the field study data described in Sections 3.3 and 3.2 will be made available online [1].

Blood glucose meter. The device is modelled with a PVS record type. The record has thirteen fields, and each field specifies either data stored in the device, such as patient account numbers, or device screens, such as "results ready". The selection of the specific type for each field is adapted to the understanding of the field investigator of the system and to the availability of information from the field study data and user manuals. In some cases details are only partial and PVS makes it possible to express partial information. An example of this can be found in the definition of patient account numbers. The observations and the user manuals reveal that they are integers, but additional details about specific constraints on the range of these numbers was not available. As a result of this the modelled patient account numbers are specified as bounded integer numbers where the bound is an uninterpreted constant (`max_patient_ID`) rather than a specific value. For other aspects, we had access to several details, but the investigator deemed it sufficient that an abstract view was given as additional information was not contributing to a better understanding of the situation. An example of this is the screen shown by the device when the clinician needs to unlock it – the specific content of the screen is not relevant to an understanding of the work. This is captured in the specification. The information is abstracted into a Boolean field, `session_unlocked`, that keeps only a high-level view about that particular device screen. Consistency constraints can be embedded as subtypes at this stage, e.g., the device must be powered on before messages are shown on the device display. The utility of embedding these constraints will be discussed further in Section 5. An excerpt from the type definition developed for the blood glucose meter follows — the relation of each field in the record type with the actual device is made clear with a comment in the specification, which is the text following the '`%`' symbol.

```
max_patient_ID: posnat  % uninterpreted constant
patient_ID        : TYPE = below(max_patient_ID)
Blood_glucose_meter: TYPE =
[# powered_on : boolean,                % ON/OFF Led
   patient_IDs: [patient_ID -> boolean], % patient IDs in the device
   quality_check_passed : boolean,      % periodic quality test screen
   session_unlocked: {b: boolean | powered_on}, % unlock screen
   ready_to_analyse: {b: boolean | powered_on}, % test ready screen
   results_ready   : {b: boolean | powered_on}, % results ready screen
   %... more device screens omitted
   result_memory_full   : boolean       % meter memory (results)
   comment_memory_full  : boolean  #]   % meter memory (comments)
```

Accessory box. The accessory box is modelled with a PVS record type. The record has six fields. Two fields are bounded natural numbers modelling white swabs and disposable lancets that are available in the box, and the other four fields are data types representing information about the test strip vials and the control solutions in the accessory box. The initial data from the field study did not provide information about the maximum swabs and lancets that can be placed in the box. The level of accuracy with which clinicians checked whether they were sufficient for visiting the beds was also not clear. Therefore we had to make decisions about the level of granularity for the models. A sensible solution was to use two symbolic constants (`n_white_swabs` and `n_lancets`), which made it possible to reason about the consequences of any possible situation during the analysis phase. In these specific constants only one constraint about the possible range values is embedded, that they are strictly greater than zero. It was known from the field investigation that the clinician (at least) visually checks whether the accessory box contains swabs and lancets before starting the visit. For the test strip vials, we modelled the following information: the barcode on the vial, specified as a new PVS abstract data type with two constructors, one constructor for readable barcodes, and the another one for unreadable barcodes; the number of strips in the vials, specified as a bounded natural number lower than a symbolic constant `n_strips`; the strip "lot", an enumerated field that specifies whether information about a set of strips is valid, invalid, expired, unknown, or not available.

```
Accessory_box: TYPE =
  [# white_swabs          : upto(n_white_swabs),
     lancets              : upto(n_lancets),
     test_strip_vial_1    : Test_strip_vial_box,
     test_strip_vial_2    : Test_strip_vial_box,
     high_control_solution: Control_solution,
     low_control_solution : Control_solution #]
```

Trolley. The trolley is modelled as a PVS record type with four fields: `cardboard`, which describes the content of the cardboard (whether it contains used gloves / swabs / other waste, and whether a checklist of beds has been annotated on the board); `glovebox`, which abstracts the glovebox as a number representing the gloves left in the box; `accessory_box`, of type `Accessory_box` defined above; `yellow_bin`, which models the content of the special refuse bin as an enumerated field whose possible values are `full`, `not_full`, `empty`, `NA`. The modelling choice for the bin was driven by the requirement to model information resources observable by the clinician. The bin is opaque, and therefore a simple visual inspection from outside is not sufficient to determine its content.

```
Trolley: TYPE =
  [# cardboard    : Cardboard,
     gloves_box   : Disposable_gloves_box,
     yellow_bin   : Special_refuse_bin,
     accessory_box: Accessory_box #]
```

Clinicians. Observable information resources are carried by clinicians during bed visits: whether they carry handover sheets, modelled as a Boolean field (`has_handover_sheets`); the actual content of the handover sheets, modelled with a field (`handover_sheets`) given as a function that maps beds in the ward to patient records; whether they have a pen or a marker for updating sheets and bed checklist, modelled as a Boolean field (`has_pen_or_marker`); whether their badge makes it possible to unlock the blood glucose meter (`has_unlocking_code`).

```
Clinician: TYPE =
  [# has_handover_sheets: boolean,
     handover_sheets    : [upto(n_beds) -> Patient_record],
     has_pen_or_marker  : boolean,
     has_unlocking_code : boolean #]
```

Patients. Observable information resources carried by patients while staying in the ward include: the wristband code, modelled as a natural number (`wristband_code`) — we chose to use a natural number rather than the type `patient_ID` defined for the blood glucose monitor model because the facts collected by the investigator did not provide any evidence that the wristband codes were generated with a system compliant with such value constraints; whether the wristband can be scanned is modelled with a Boolean field (`wristband_can_be_scanned`) — sometimes the barcode cannot be scanned because, for instance, the wristband has been accidentally crumpled; the patient name (`patient_name`), a string storing the patient name — we are not including any assumption about whether the names are unique, as the investigation did not provide such evidence; the bed number (`bed_number`), a bounded integer number identifying one of the beds in the ward.

```
Patient: TYPE =
  [# wristband_code: nat,
     wristband_can_be_scanned: boolean,
     patient_name  : string,
     bed_number    : upto(n_beds) #]
```

Activities. The clinicians are involved in activities that are specified as transition functions over system states defined in terms of the data types described in the previous section. The state of the system collects together, possibly multiple instances, of the system elements: trolley, device, clinician, and patient. Since the aim of modelling activity is to describe precisely the observed use of artefacts and information resources to determine whether there are gaps or mismatches, an activity is defined for each step described in Section 3.3. An illustration of the formalisation of the activity is `puncture_finger`, which specifies the use of artefacts on the trolley during the first phases of the blood glucose monitoring test. The aim of the illustration is to give the reader a taste of the modelling style and of the decisions that need to be made during the modelling process. The formalisation of the other activities will be made available online [1].

From the field study data, the following facts about the use of artefacts and information resources when puncturing the patient's finger to get a blood sample and start the test can be discerned.

- The clinician uses at least one lancet and one white swab for each test. A precise estimate of the total number of lancets and swabs that will be used in each test cannot be identified in advance, because the test might need to be repeated, e.g., because the alcohol used to cleanse the puncture site may lead to error codes or inaccurate reading, or because the collected blood sample was not sufficient or was excessive. This uncertainty is modelled using two symbolic constants (`used_lancets` and `used_swabs`) in the arithmetic expression for updating the number of lancets and white swabs in the accessory box. PVS is able to generate proof obligations using these expressions that enables exploration of what may happen for different ranges of values for the symbolic constants. An illustration is provided in Section 5.
- The clinician may use a cardboard container that is either empty or contains used items, e.g., because the test had to be repeated more than once, or because the clinician is delaying the transfer of used items to the bin. This uncertainty can be expressed using a random Boolean variable[3].
- The clinician needs to use disposable gloves. The number of gloves needed cannot be estimated in advance in a precise way because of the particular work environment. For instance, clinicians may be temporarily interrupted during the blood glucose monitoring test because they need to carry out another task. As for the number of lancets and white swabs, we therefore use an arithmetic expression with a symbolic constant `used_gloves`.

```
puncture_finger(sys: State): State =
sys WITH [ trolley := trolley(sys)
    WITH [ accessory_box := trolley(sys)‘accessory_box
    WITH [ lancets := trolley(sys)‘accessory_box‘lancets - used_lancets,
           white_swabs := trolley(sys)‘accessory_box‘white_swabs
                                                    - used_swabs,
           cardboard := trolley(sys)‘cardboard
              WITH [ used_gloves := choose(fullset[boolean]),
                     used_swabs  := choose(fullset[boolean]) ],
           gloves_box := trolley(sys)‘gloves_box - used_gloves ]]]
```

5 Using PVS to support the field investigation: analysis and results

The modelling exercise was itself a first analysis step. The modelling approach proved useful as a means of helping the field investigator organise a substantial number of field notes, which were frequently incomplete and interleaved with

[3] In PVS, random values for any datatype can be obtained with the `choose` function, which takes as argument the set from which the random element must be chosen.

other observations of devices. The modelling process achieved clarification and made the subsequent stages of the analysis faster and more effective.

5.1 Issues emerged while developing the PVS models

Several questions warranting further investigation as part of the field study were raised while developing the models. Some relevant examples follow.

Quality control. The aim of the quality control performed when scanning the barcode on the test strip vials is to calibrate the device to a specific set of strips and to check whether the strips are valid and approved. Modelling this information from the user manual stimulated further investigation of subtle inconsistencies in the mental models possessed by clinicians. For instance, some clinicians believed that the test strip scanning was only for checking the validity of the strips. This understanding is correct but partial, and opens the possibility of calibrating the meter for a vial of test strips and, in the case the strip vial is empty, using strips from another uncalibrated set — the accessory box has two strip vials and the container is opaque, therefore clinicians are not able to visually check whether the vial has strips in it when scanning the barcode.

Barcode scanner. The barcode scanner embedded in the device is used to read clinician's badges and patient's wristbands. Whereas the field investigator had accepted the technology as standard, a systematic analysis of the user manuals made clear that different subtle variants of the same technologies could have been implemented in the device. Each technology could be more or less suited to the context. This also helped to clarify the origin of various minor disturbances observed during the study, e.g., when the device was not scanning correctly the barcode on the wristband.

User manual. Another advantage of the added analysis included an awareness of what was and was not in the device manual. Data solely from the field can be understated because it is assumed by participants and not particularly salient. However, when contrasted with the official and concrete account in the manual the importance of these differences is emphasised, e.g., the manual says that if there is not enough blood placed on a test strip then this can hinder the reading but staff also said that too much can also hinder the reading.

5.2 Issues raised by PVS when checking model consistency

Proof obligations automatically generated by PVS stimulated further constructive discussion after the modelling step. In the following, we illustrate some of these situations for the `puncture_finger` activity specified above in Section 4. For that activity, PVS automatically generated various proof obligations that could not be discharged using the facts available from the field study. Two examples illustrate this.

Lancets. An undischargable proof obligation automatically generated by PVS related to the arithmetic expression containing the symbolic constants. PVS requires the expression that updates the number of lancets to result in a value that is non-negative. This translates into a request to the field study to check whether the accessory box is guaranteed to contain enough lancets for the tests:

```
% Subtype TCC generated for
% trolley(sys)‘accessory_box‘lancets - used_lancets
% expected type upto(n_disposable_lancets)
puncture_finger_TCC1: OBLIGATION
 FORALL (sys: State):
    sys‘trolley‘accessory_box‘lancets - used_lancets >= 0
```

This obligation, although mathematically trivial, drew our attention to the lancets and stimulated further investigation about how they are actually used by clinicians. The investigation started as a clarification of whether they were single use, but then allowed the investigator to get useful insight about the risks of not putting them in the sharps bin straight away and also getting them mixed up with the unused lancets (it is hard to distinguish used lancets from unused ones by visual inspection). Some of these latent situations were actually observed in subsequent visits to the ward, when the trolley was not used because only a few patients needed to be checked.

Battery. A second undischargable proof obligation generated by PVS resulted from the reasonable conjecture that the device is ready for the test. The conjecture claimed that a precondition that the device be unlocked be true. We composed the modelled activities to specify a symbolic execution trace and verify the conjecture. Interestingly, PVS generates a proof obligation because of a consistency constraint imposed in the model: in order to be unlocked, the device must be also powered on:

```
% Subtype TCC generated for
% symbolic execution trace
% expected type {sys: State | device(sys)‘powered_on
%                             AND NOT device(sys)‘session_locked}
symbolic_execution_trace_TCC2: OBLIGATION
 confirm_strip(enter_patient_ID(...prepare_trolley(initial_state)...)
    ‘device‘powered_on
  AND
 confirm_strip(enter_patient_ID(...prepare_trolley(initial_state)...)
    ‘device‘session_unlocked
```

This detail has drawn attention to the fact that the device is battery powered, and that wear and tear inevitably reduces the autonomy of the device. This can be a source of disturbances during visits to several beds, because the clinician powers on the device at the beginning of the visit and keeps the device powered

on until the visit completes. In some cases, the device may therefore run out of power and require replacement with another one. This raises a question about latent situations where the trolley may be left unattended for short periods of time in order to collect another device.

6 Related work and conclusions

Little work has been done on approaches that integrate formal methods and empirical studies, and hardly any concrete examples can be found of case studies involving real world systems. This is partially linked to the complexity of using formal methods, which require a steep learning curve, and to additional barriers (some real, some imagined) created by the current generation of user interfaces for automated reasoning tools, which are essentially text-based and practically inaccessible to non-expert users.

Wright, Fields and Merriam [15] investigated the possibility of defining a conceptual framework for integrating formal methods and empirical approaches for studying interactive systems. They argue that extant artefacts and informal understanding of the system can provide insights about usability properties that might be of interest. This informal understanding can then be refined through formal methods by generating design questions and evaluating design alternatives, which can in turn be evaluated empirically, e.g., through prototypes. They demonstrate the approach with an example based on a web browser. Fields [4] applied the same approach to the analysis of a remote control system. Rukšėnas et at [11] combined empirical studies and mathematical models to formalise the relationship between salience and cognitive load revealed by lab experiments. The outcome of the lab experiments was used to refine the salience rules defined in the mathematical model, and the outcome of the model-based analysis was able to generate new experimental hypotheses for researchers in cognitive science. These works share with ours the argument that informal approaches and formal methods have complementary roles in the analysis of the system.

In our previous works, we have shown that the integrated analysis proposed by Wright, Fields and Merriam can be extended to the wider socio-technical system. We demonstrated the utility of the approach when re-analysing field study data from already completed field investigations performed in different contexts and for different aims, including: use of drug infusion pump in a day care unit [2], analysis of an emergency dispatch system [6,7], and analysis of incidents reports [5,8].

In this work, we have explored the possibility of using the PVS verification tool to support an investigator while conducting a field study. For the developed models, PVS is able to generate proof obligations and proof attempts in seconds. We have provided some evidence of the utility of the approach. As gaps and inconsistencies are uncovered within and between the various specifications, new questions are generated, which can be used to refine the field study. Also, the formal specification can be refined as new facts are discovered — the two methods feed each other.

Acknowledgements. CHI+MED: Multidisciplinary Computer-Human Interaction research for the design and safe use of interactive medical devices project, EPSRC Grant Number EP/G059063/1. Extreme Reasoning, Grant Number EP/F02309X/1.

References

1. PVS models of field study at oncology ward, June 2012. PVS models available online at `http://tinyurl.com/PVS-bloodglucosestudy`.
2. A. Blandford, A. Cauchi, P. Curzon, P. Eslambolchilar, D. Furniss, A. Gimblett, H. Huang, P. Lee, Y. Li, P. Masci, P. Oladimeji, A. Rajkomar, R. Rukšėnas, and H. Thimbleby. Comparing actual practice and user manuals: A case study based on programmable infusion pumps. In *Eics4Med, the 1st Intl. Workshop on Engineering Interactive Computing Systems for Medicine and Health Care*, pages 59–64. ACM Digital Library, 2011.
3. A. Blandford and D. Furniss. DiCoT: A Methodology for Applying Distributed Cognition to the Design of Teamworking Systems. *Interactive Systems*, pages 26–38, 2006.
4. R. Fields. *Analysis of Erroneous Actions in the Design of Critical Systems.* PhD thesis, University of York, 2001.
5. P. Masci and P. Curzon. Checking user-centred design principles in distributed cognition models: a case study in the healthcare domain. In *USAB 2011: Information Quality in eHealth, 7th Conference of the Austrian Computer Society*, pages 95–108. Springer Lecture Notes in Computer Science (LNCS), 2011.
6. P. Masci, P. Curzon, A. Blandford, and D. Furniss. Modelling distributed cognition systems in PVS. *ECEASST*, 45, 2011.
7. P. Masci, P. Curzon, D. Furniss, and A. Blandford. Using PVS to support the analysis of distributed cognition systems. *Submitted for publication to Innovations in Systems and Software Engineering*, 2012.
8. P. Masci, H. Huang, P. Curzon, and M.D. Harrison. Using PVS to investigate incidents through the lens of distributed cognition. In Alwyn E. Goodloe and Suzette Person, editors, *Proceedings of the 4th NASA Formal Methods Symposium*, volume 7226, pages 273–278, Berlin, Heidelberg, April 2012. Springer-Verlag.
9. Roche. Accu-Chek Inform II System, Professional glucose testing for the wireless age, August 2010.
10. Roche. Accu-Chek Inform System, Operator's manual, November 2010.
11. R. Rukšėnas, J. Back, P. Curzon, and A. Blandford. Verification-guided modelling of salience and cognitive load. *Formal Aspects of Computing*, 21(6):541–569, 2009.
12. N. Shankar and S. Owre. Principles and pragmatics of subtyping in PVS. In D. Bert, C. Choppy, and P. D. Mosses, editors, *Proc. of WADT '99*, volume 1827 of *Lecture Notes in Computer Science*, pages 37–52. Springer-Verlag, 1999.
13. NHS Trust. Accu-Chek Inform II Blood Glucose Meter Standard Operating Procedure, August 2011.
14. J.I. Westbrook and A. Ampt. Design, application and testing of the Work Observation Method by Activity Timing (WOMBAT) to measure clinicians' patterns of work and communication. *International Journal of Medical Informatics*, 78, 2009.
15. P.C. Wright, R. Fields, and N.A. Merriam. *From formal models to empirical evaluation and back again*, chapter 13, pages 283–314. Formal methods in human-computer interaction. Berlin, Springer, 1997.